

Rejuvenating a Classic

Jim Springfield
Principal SDE

Mark Hall
Principal SDE

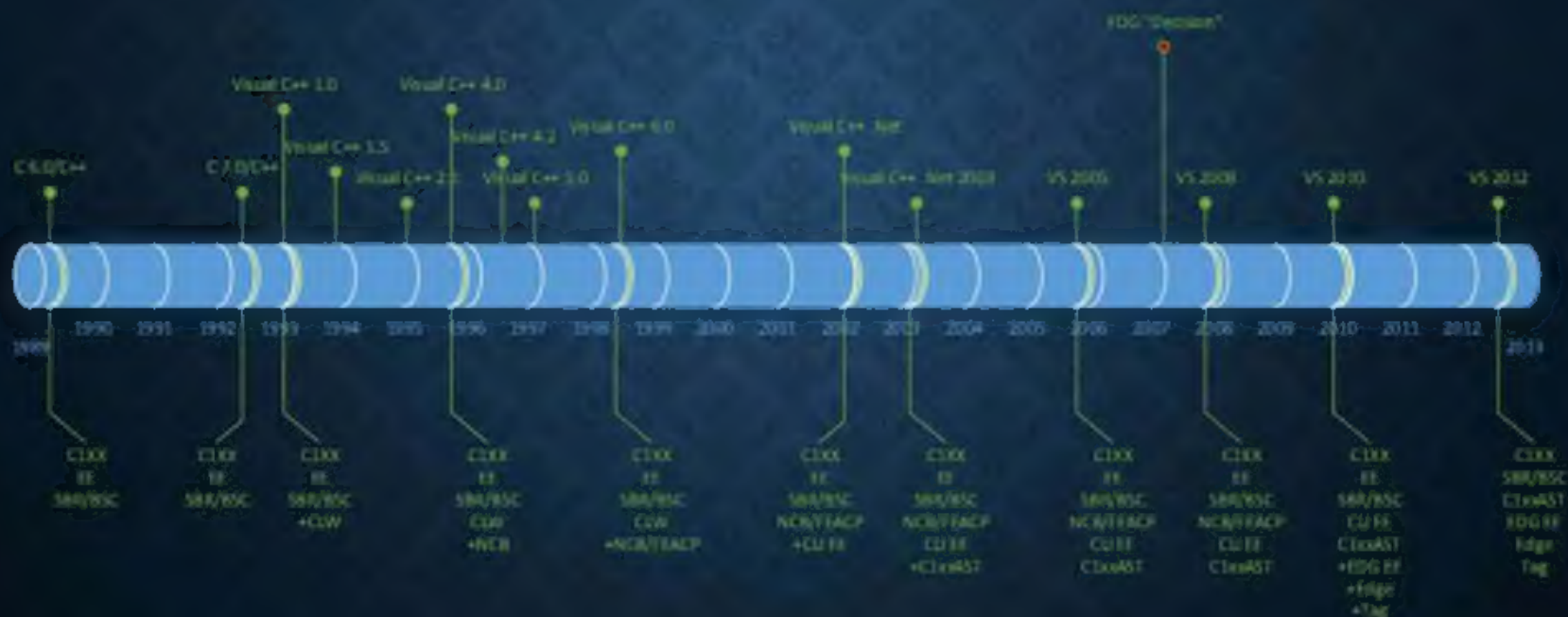


REJUVENATING A CLASSIC





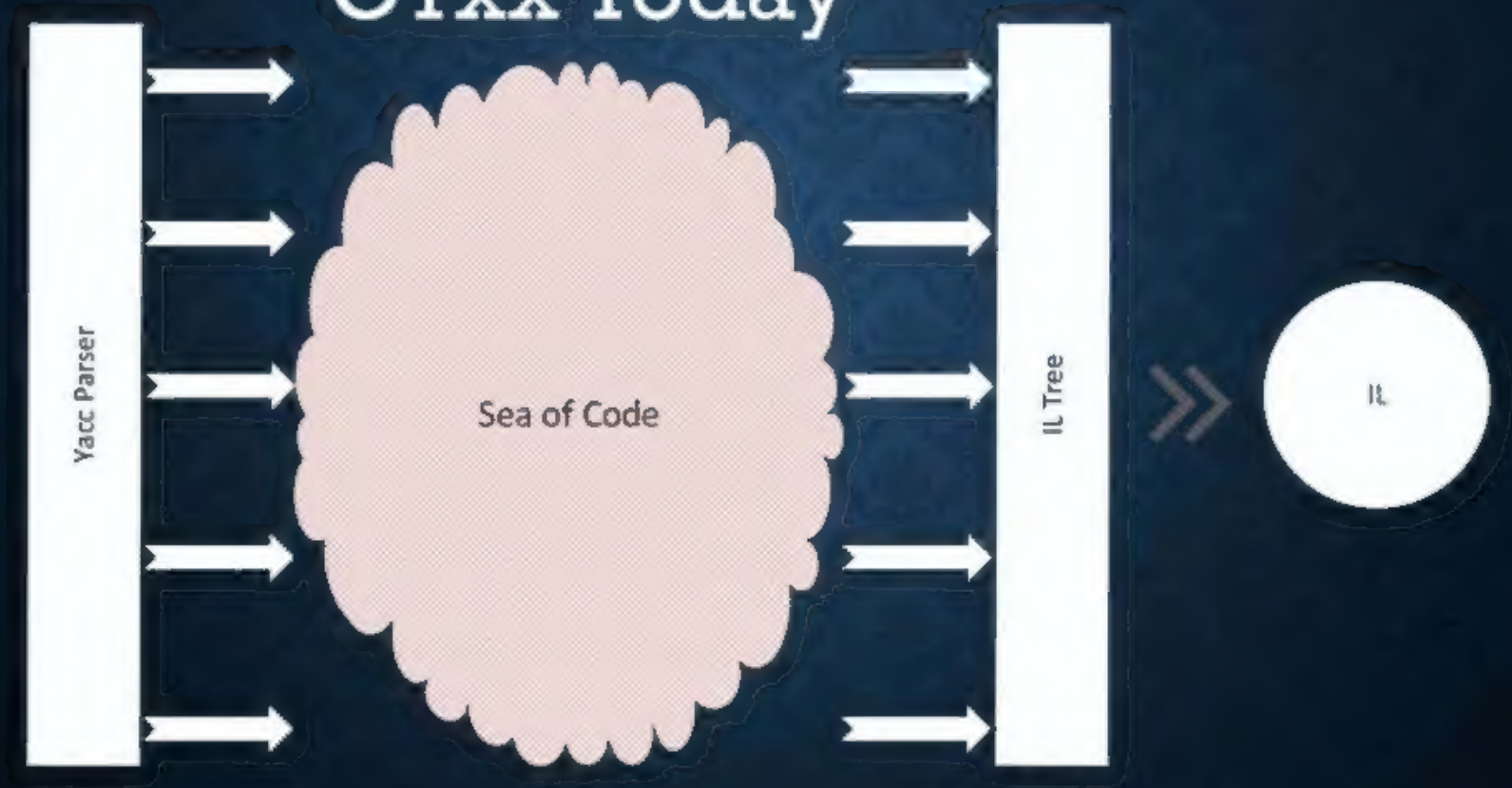
A BRIEF HISTORY OF MICROSOFT C++



C++ Parser Use	Binary	CodeBase	Products	Notes	PP	Lex	Parse	AST	Sem	IL	Other
Batch Compiler	clxx.dll	C1XX	VC1-		X	X	X		X	X	
Prefast (/analyze)	clxxast.dll	C1XX (#if)			X	X	X	X	X		Plugin
FEACP Intellisense	feacp.dll	C1XX (#if)	VC8-2008	No longer used	X	X	X		X		
Edge Intellisense	cpfe.dll	EDG + Extensions	Dev10-		X	X	X	X	X		
Native EE		Custom	VC1-	Deprecated		X	X	X	X		Eval
Managed EE		Custom	2003-	Deprecated		X	X	X	X		Eval
Concord EE		EDG+Ext	Dev11-			X	X	X	X		Eval
Tag Parser	cpfe.dll	ANTLR + EDG Lexer	Dev10-	Browsing	X	X	X	X			
IDE Lexer	vcpkg.dll	Custom	VC8-	Color/BM	min	X	min				
CodeDom	vcpkg.dll	Custom	2003-	WinForms		X	X	X			
CodeModel	vcpkg.dll	Custom	2003-	Misc		X	X	X			

PP – Preprocessor | Lex – Lexer | Parse – Parser | AST – Abstract Syntax Tree generation | Sem – Semantic Analysis | IL – IL generation

C1xx Today



TENETS

- Stay testable
 - And stay tested
- Use compiler to verify itself
 - Improve type safety of codebase
 - Use tree/CIL comparison to prevent accidental diffs
- Preserve value where it exists
 - Avoid rewriting where possible/Arrange existing code into phases
- Lazy compilation (Just-In-Time)
 - Don't do work too early (constant folding, high-low transformations)
- Breadth-first
 - Don't add to overall complexity
 - Don't introduce new mechanisms without removing old mechanisms
 - Realize benefits from each milestone even if project is cancelled/suspended
- Remove preprocessor/build variations
 - Single binary for batch/AST/Intellisense, 32/64 bit targeting
 - Consider hosting tools only on x64

RISK MITIGATION

- Performance
 - By remaining testable and checking in to the feature branch any perf issues can be discovered quickly
- Boiling the ocean
 - Take the smallest step possible, but no smaller
- Management support
 - Provide continuous value

TIMELINE



- Ship analysis replacement in Dev14.
- Start Intellisense work once CA is wrapped up (late 2014 – early 2015).
- No plan for replacing EE (expression evaluator) yet.